B4 end

16. The circuit of claim 15 wherein said timing means includes means for varying the rate of change of said priority levels as a function of time.

REMARKS

The claims have now been amended to clarify aspects of the invention and to indicate with more definiteness the nature of the invention claimed. For example, claim 2 has been amended to recite the step of providing an integrated circuit for automating the selection of tasks to be run by the computer system. This is to distinguish the invention from task management systems which are procedural methods carried out manually or by software. Claim 2 makes it clear that the method steps in paragraph (b) are carried out by a circuit specially designed for this purpose.

The examiner rejected claim 2 under 35 USC §112 because of a misunderstanding regarding the incrementing function. All tasks have a priority level assigned to them based upon some parameter during task initialization. As the specification indicates, all time critical tasks have their priorities changed as a function of time and at possibly different rates as priority values change at different speeds. Since only the highest priority task is allowed to run, the purpose of the time incrementing function is to ensure that each task's priority level achieves a sufficiently high value to run within it allotted period of time. This insures that all ready-to-run tasks will achieve a high enough priority to run. In other words, after a finite period of time, the priority of the task that must be run will eventually increase to the point at which said task has the highest priority and will indeed run.

Claim 7 presents a different case. In claim 7, the timer function may be implemented by a timeout counter register (91) which generates an interrupt signal through gate 99. Thus, one of the ways that a task is designated "ready-to-run" is when its timeout register value

reaches a set value and outputs a signal to the interrupt control 96. The amendments to claim 7 are believed to make this relationship clear.

Claim 8 was rejected under §112 because of an apparent lack of connection between the task switching and task tracing functions recited. Thus, the claim has been amended to make clear that the task tracing function is provided in addition to task switching. The apparent lack of antecedent basis for the element "register states" is not understood. This is the first occurrence of this element and thus there is no antecedent basis required. The claim does not say "the register states" or "said register states" — it merely says "register states".

Claim 9 has been amended to make clear the relationship between the task selection circuit and the zero overhead switching of tasks. Thus, the zero overhead switching (multiplexing) circuit is coupled to the output of the task change processing circuit. In other words, the selected task and the next ready to run task are placed in latches so that they may be multiplexed when the running task finishes or is interrupted. Thus, switching between tasks is accomplished with zero overhead, "overhead" being defined as wasted clock cycles that would ordinarily be consumed with the fetching and evaluation of other tasks. The invention automates this function in hardware using the claimed circuitry.

The claims were rejected as obvious over Madnick, either alone or in combination with George '691 or Jen '365. The rejections are, respectfully, traversed by applicant. Madnick is a textbook written in 1974. It describes procedures and protocols for running various tasks on a data processing system. These procedures are carried out manually for the most part using a set of rules to identify tasks by priority. Tasks in these systems frequently consisted of program instructions coded in stacks of punch cards that were placed in bins and sorted by hand according to a set of priority rules. The cards were then loaded into the computer, usually by hand, and processed by the computer. There was no circuit or other piece of hardware capable of automatically determining the order of priority of a plurality of tasks that in

the claimed invention come from a multiplicity of sources such as hardware interrupts that constantly request changes in the order of tasks to be run. In contrast to Madnick, the invention accomplishes this with a circuit formed by a collection of logic elements. Madnick teaches no such circuit nor does it even hint that such technology could even be possible. This is not surprising given Madnick's very early publishing date. The microcomputer had not been invented yet.

Thus, claim 2, which has been amended to recite that the process steps are carried out by a circuit, patentably distinguishes over Madnick. Likewise, claim 4, which states in its preamble that it is an interrupt and task change processing circuit, is neither taught nor suggested by anything in Madnick. The concepts and procedures described in Madnick have little to do with modern real time operating systems and any similarity between the two has more to do with the fact that similar words are used to describe very different functions. Madnick does not describe or even remotely hint at a system in which a set of pre-initialized but autonomously executing circuit can control dynamically and in real time a sequence of tasks to be executed by a computer. Claims 2 and 4 are thus patentable over Madnick and should be allowed.

Claims 5-9 are dependent on claim 4 and are thus patentable over the art of record. Claims 14-16 are dependent on claim 2 and are therefore patentable. New claims 10-13 are presented for examination and are believed to be patentably distinct over the art of record.

Respectfully submitted,

William O. Geny

COPY OF PAPERS ORIGINALLY FILED

Applicant:	Brian DONOVAN	Group Art Unit:	2783
Serial Nº:	09/410,202	Examiner:	D. Eng
Filed:	September 30, 1999		
Title:	ZERO OVERHEAD COMPUTER INTERRUPTS WITH TASK SWITCHING		

Deletions are bracketed; additions are underlined.

The following material shows the manner of amendment of the specification.

2. A method for ordering the performance of tasks in a computer system, said computer system having input sources that issue interrupt signals for requesting the performance of a task, said method comprising:

(a) [maintaining] providing an integrated circuit having circuit components for automating the selection of tasks to be performed by said computer system;

(b) wherein the integrated circuit performs the following steps:

(i) assigning a priority level for each task based upon a selected parameter of an interrupt signal;

(ii) [incrementing] changing each priority level as a function of time; and

([c]iii) beginning the execution of a first task when said priority level of said first task exceeds said priority level of all other tasks.

4. In a [data processing] microprocessor-based computing system having a CPU for executing tasks represented by task register sets and further including peripheral devices that issue interrupt commands, an interrupt and task change processing circuit comprising:

(a) a task enable circuit for determining from predetermined inputs whether a predetermined task is ready for execution by [a] the central processing unit,

(b) a task priority selection circuit coupled to an output of the task enable circuit for determining an order for the running of tasks

that have been determined ready for execution by the task enable circuit; and

(c) a task switching circuit coupled to an output of the task priority selection circuit for controlling the execution of tasks in a sequence determined by the task priority selection circuit.

6. The interrupt and task change processing circuit of claim 4 wherein the task enable circuit includes a task linking circuit for linking together [a plurality] groups of tasks [in a predetermined] which are dependent upon each [order] other.

7. The interrupt and task change processing circuit of claim 4 wherein the task enable circuit [includes] is responsive to a task interrupt signal for designating a task as ready to run, said task enable circuit including a timer for generating [a] said task interrupt signal after a predetermined period of time.

8. The interrupt and task change processing circuit of claim 4 [wherein the task switching circuit includes] further including a trace enable circuit for recording register states [on] of selected registers during a preselected clock cycle.

9. The interrupt and task change processing circuit of claim 4 wherein the task switching circuit [includes] is coupled to a zero overhead multiplexing circuit for storing a later task in a first set of latches during a first clock cycle while simultaneously switching a previously stored earlier task stored in a second set of latches into a task switch controller during the same clock cycle.

COPY OF PAPERS
ORIGINALLY FILED

CERTIFICATE OF MAILING

I hereby certify that this Preliminary Amendment is being deposited with the United States Postal Service as first class mail on May 10, 2002 in an envelope addressed to: BOX DAC, Office of Petitions, Assistant Commissioner for Patents, Washington, D.C. 20231.

Dated: May 10, 2002

William O. Geny